**[String Conversion and toString() in Java](http://www.java-samples.com/showtutorial.php?tutorialid=218)**

When Java converts data into its string representation during concatenation, it does so by calling one of the overloaded versions of the string conversion method **valueOf( )** defined by **String**. **valueOf( )** is overloaded for all the simple types and for type **Object**. For the simple types, **valueOf( )** returns a string that contains the human-readable equivalent of the value with which it is called. For objects, **valueOf( )** calls the **toString( )** method on the object. We will look more closely at **valueOf( )** later in this chapter. Here, let's examine the **toString( )** method, because it is the means by which you can determine the string representation for objects of classes that you create.

Every class implements **toString( )** because it is defined by **Object**. However, the default implementation of **toString( )** is seldom sufficient. For most important classes that you create, you will want to override **toString( )** and provide your own string representations. Fortunately, this is easy to do. The **toString( )** method has this general form:

String toString( )

To implement **toString( )**, simply return a **String** object that contains the human-readable string that appropriately describes an object of your class.

By overriding **toString( )** for classes that you create, you allow the resulting strings to be fully integrated into Java's programming environment. For example, they can be used in **print( )** and **println( )** statements and in concatenation expressions. The following program demonstrates this by overriding **toString( )** for the **Box** class:

**// Override toString() for Box class.   
class Box {   
double width;   
double height;   
double depth;   
Box(double w, double h, double d) {   
width = w;   
height = h;   
depth = d;   
}   
public String toString() {   
return "Dimensions are " + width + " by " +   
depth + " by " + height + ".";   
}   
}   
class toStringDemo {   
public static void main(String args[]) {   
Box b = new Box(10, 12, 14);   
String s = "Box b: " + b; // concatenate Box object   
System.out.println(b); // convert Box to string   
System.out.println(s);   
}   
}**

The output of this program is shown here:

Dimensions are 10 by 14 by 12.   
Box b: Dimensions are 10 by 14 by 12.

**Note**

**As you can see, Box's toString( ) method is automatically invoked when a Box object is used in a concatenation expression or in a call to println( ).**